

Postmortem Menú Reporte Materias

25/04/17

0.1

|  |  |
| --- | --- |
| **Nombre.** | **Matricula** |
| **Jesús Alberto Goiz Barrales.** | **24400085** |
| **Juan Alberto Gutiérrez canto.** | **24400063** |
| **Brenda Robles Antonio.** | **24400073** |
| **Rene Moratilla Montes.** | **24400075** |
| **Guillermo Vivaldo Vazquez** | **24400093** |

**Control De Versiones**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre Del Archivo | Versión | Fecha | Autor | Comentarios |
| Csoft-PM\_Menú\_Reporte\_Materias-250417.docx | 0.1 | 25/05/17 | RMM | Creación del documento |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Índice

[**Índice**](#_kd5bfhkalq0n) **2**

[**I.- Forma PSP Project Plan Summary.**](#_jhy44tc5k06j) **3**

[**II.- Time Recording Log.**](#_1fob9te) **3**

[**III.- Defect Recording Log.**](#_3znysh7) **3**

[**IV.- Diseño.**](#_m4hw86ggwsmr) **3**

[**V.- Design CheckList.**](#_92tu2pewgguj) **3**

[**VI.- Programa Fuente.**](#_6bvnqqrwimi9) **4**

[**VII.- Code CheckList.**](#_87ns93g4124c) **4**

[**VIII.- Reporte de resultados.**](#_bo7s7l2hg3ra) **4**

[**IX.- PIP**](#_4ocjjwajumc0) **4**

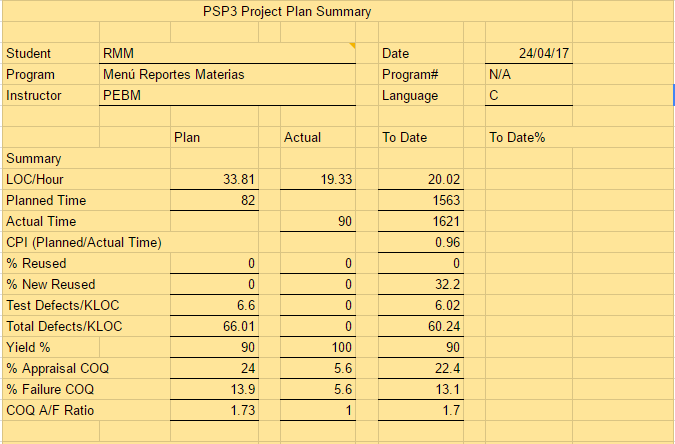
[**X.- Size Estimating Template**](#_w1ych8jaj7as) **5**

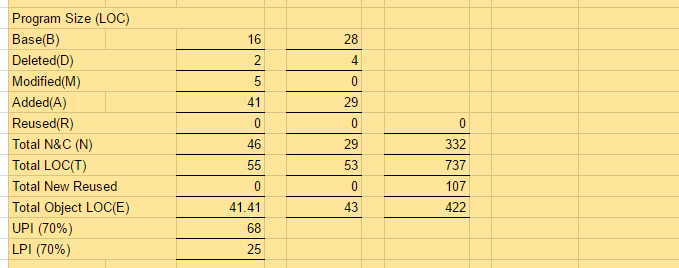
[**XI.- Task and Schedule**](#_195jonwtvlxs) **5**

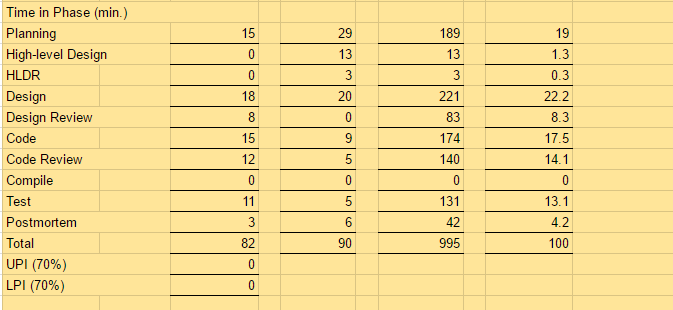
# 

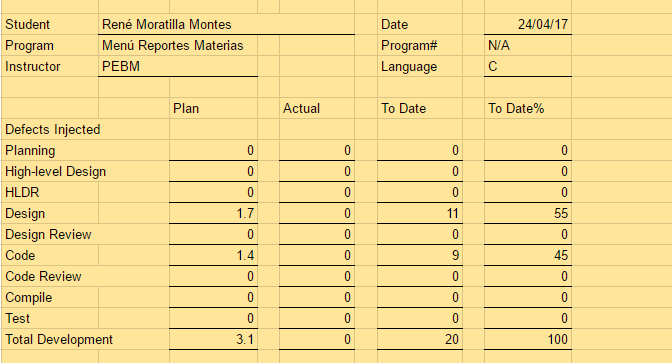
# 

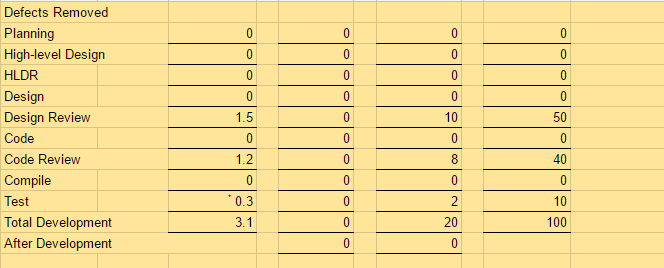
# I.- Forma PSP Project Plan Summary.

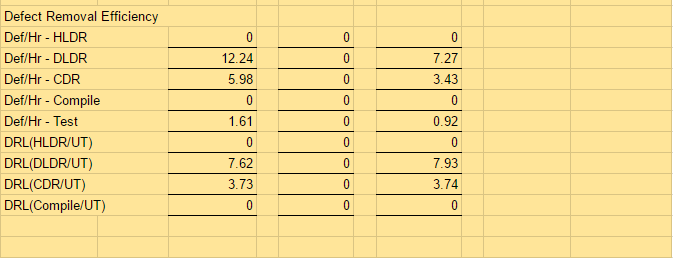




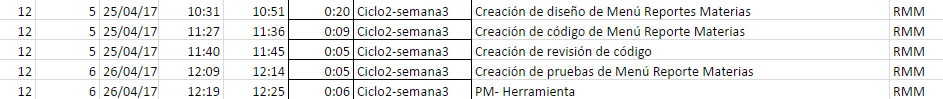








# II.- Time Recording Log.



# III.- Defect Recording Log.



# IV.- Diseño.

# Functional Specification Template

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Student** | | | RMM | | **Date** | 25/04/17 |
| **Program** | | | Menú Reportes MAterias | | **Program #** | N/A |
| **Instructor** | | | PEBM | | **Language** | C |
|  | | | | | | |
| **Class Name** | | void menu\_reportes\_materias() | | | | |
| **Parent Class** | | N/A | | | | |
|  | | | | | | |
| **Attributes** | | | | | | |
|  | **Declaration** | | | **Description** | | |
|  | Int opcion | | | Corresponde a la opción que el usuario eligió. | | |
|  | | | | | | |
| **Items** | | | | | | |
|  | **Declaration** | | | **Description** | | |
|  | reporte\_materias\_semestres() | | | Presenta el vinculó a el reporte materias por semestre | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |

# Logic Specification Template

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Student | RMM | | Date | 25/04/17 |
| Program | Menú Reporte Materias | | Program # | N/A |
| Instructor | PEBM | | Language | C |
| Object | N/A | | Function | menu\_reportes\_materias() |
| **INCLUDES:**  **TYPE DEFINITIONS:**  **Declaration:** | | #include <iostream>  #include <cstdio>  #include <cstdlib>  using namespace std; | | |
|  | |  | | |

|  |  |
| --- | --- |
| **Reference:** |  |

|  |  |
| --- | --- |
| Logic reference numbers | Program logic, in pseudocode |
| 1 | Imprime el menú reportes materias con todas sus opciones |
|  | Leer la opción del usuario |
|  | En caso de que sea 1 |
|  | Va a la función reporte\_materias\_semestre() |
|  | En caso de que sea 2 |
|  | Regreso a el menú de reportes |
|  | En caso de que sea otro carácter |
|  | Imprime opción incorrecta |
|  | fin. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Operational Specification Template

|  |  |  |  |
| --- | --- | --- | --- |
| **Student** | RMM | **Date** | 25/04/17 |
| **Program** | Menú Reporte Materias | **Program #** | N/A |
| **Instructor** | PEBM | **Language** | C |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scenario Number** | **1** | **User Objective** | Seleccione ir a reportes materias por semestre | |
| **Scenario Objective** | | Pasos para ver el reportes materias por semestre | | |
| **Source** | **Step** | **Action** | | **Comments** |
| Usuario | 1 | Abril el programa. | |  |
| Programa | 2 | Desplegar el menú reportes materias | | Dando las opciones de los procesos siguientes. |
| Usuario | 3 | Ver el menú reportes materias | | Checar las opciones posibles |
| Usuario | 4 | Introducir opción deseada 1 | | Elegir entre las opciones posibles |
| Programa | 5 | Leer opción del usuario | | Verificar que sea valida |
| Programa | 6 | Mandar a el reporte de materias por semestre | | Dirige a la función de reportes\_materias\_semestres() |
| Usuario | 8 | fin | | Presentar lo del módulo materias. |
|  |  |  | |  |
|  |  |  | |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scenario Number** | **2** | **User Objective** | Seleccione ir a menú de reportes | |
| **Scenario Objective** | | Pasos para ver el menú de reportes | | |
| **Source** | **Step** | **Action** | | **Comments** |
| Usuario | 1 | Abril el programa. | |  |
| Programa | 2 | Desplegar el menú reportes materias | | Dando las opciones de los procesos siguientes. |
| Usuario | 3 | Ver el menú reportes materias | | Checar las opciones posibles |
| Usuario | 4 | Introducir opción deseada 2 | | Elegir entre las opciones posibles |
| Programa | 5 | Leer opción del usuario | | Verificar que sea valida |
| Programa | 6 | Mandar a el menú de reporte | | Dirige a la función de menu\_reportes() |
| Usuario | 8 | fin | | Presentar lo del módulo menú de reportes.. |
|  |  |  | |  |

# State Specification Template

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Student | | RMM | | | Date | | 25/04/17 |
| Program | | Menú Reporte Materias | | | Program # | | N/A |
| Instructor | | PEBM | | | Language | | C |
|  | | |  | | | | |
| Menú principal | | | | | | | |
| **State Name** | | | | **Description** | | | |
| Inicio | | | | Inicia el programa. | | | |
| Menú reportes materias | | | | Se presentará el menú reportes de materias | | | |
| Salir | | | | Termina el programa | | | |
|  | | | |  | | | |
| **Function/Parameter** | | | | **Description** | | | |
| printf() | | | | Imprime datos en la pantalla | | | |
| scanf() | | | | Lee datos de la pantalla | | | |
| opción | | | | opción que elegirá el usuario | | | |
| reporte\_materias\_semestres() | | | | Presenta el reporte de alumnos materias | | | |
| **States/Next States** | | | | **Transition Condition** | | **Action** | |
| Inicio | | | |  | |  | |
|  | Menú principal | | | True | | printf()-imprime el menu  scanf()- lee opción | |
| Menú principal | | | |  | |  | |
|  | Materias por Semestres | | | opción=1 | | reporte\_materias\_semestres() | |
|  | Salir | | | opción=2 | | Fin. | |

# V.- Design CheckList.

**PSP2 Design Review Checklist**

|  |  |  |  |
| --- | --- | --- | --- |
| Student | RMM | Date | 25/04/17 |
| Program | Menú Reporte Materias | Program # | N/A |
| Instructor | PEBM | Language | C |

|  |  |
| --- | --- |
| Purpose | To guide you in conducting an effective design review |
| General | - Review the entire program for each checklist category; do not attempt to review for more than one category at a time!  - As you complete each review step, check off that item in the box at the right.  - Complete the checklist for one program or program unit before reviewing the next. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Complete | Verify that the design covers all of the applicable requirements.  - All specified outputs are produced.  - All needed inputs are furnished.  - All required includes are stated. | MB  X  X  X | B | R | M |
| External Limits | Where the design assumes or relies upon external limits, determine if behavior is correct at nominal values, at limits, and beyond limits. |  |  |  |  |
| Logic | - Verify that program sequencing is proper.  Stacks, lists, and so on are in the proper order.  Recursion unwinds properly.  - Verify that all loops are properly initiated, incremented, and terminated.  - Examine each conditional statement and verify all cases. | X    X    X |  |  |  |
| Internal Limits | Where the design assumes or relies upon internal limits, determine if behavior is correct at nominal values, at limits, and beyond limits. |  |  |  |  |
| Special Cases | - Check all special cases.  - Ensure proper operation with empty, full, minimum, maximum, negative, and cero values for all variables.  - Protect against out-of-limits, overflow, and underflow conditions.  - Ensure “impossible” conditions are absolutely impossible.  - Handle all possible incorrect or error conditions. | X  X    X  X  X |  |  |  |
| Functional Use | - Verify that all functions, procedures, or methods are fully understood and properly used.  - Verify that all externally referenced abstractions are precisely defined. | X    X |  |  |  |
| System Considerations | - Verify that the program does not cause system limits to be exceeded.  - Verify that all security-sensitive data are from trusted sources.  - Verify that all safety conditions conform to the safety specifications. | X    X    X |  |  |  |
| Names | Verify that  - all special names are clear, defined, and authenticated  - the scopes of all variables and parameters are self-evident or defined  - all named items are used within their declared scopes | X  X  X |  |  |  |
| Standards | Ensure that the design conforms to all applicable design standards. | X |  |  |  |

# VI.- Programa Fuente.

/\*=========================================================\*/

/\*Autor: René Moratilla Montes \*/

/\*Componente: Menú Reporte Materias \*/

/\*Fecha: 25/04/2017 \*/

/\*Descripción: Codificación que muestra el menú reporte materias\*/

/\*Versión: 0.1 \*/

/\*=========================================================\*/

/\*==========================================================\*/

/\* El programa contiene lo siguiente: \*/

/\* void reporte\_materias\_semestres() \*/

/\* void menu\_reporte\_materias() \*/

/\* int main() \*/

/\*==========================================================\*/

#include <iostream>

#include <cstdio>

#include <cstdlib>

using namespace std;

/\*\*\*Esta función se encarga de presentar el menú principal y redireccionar a la opcion que el usuario eligio\*\*\*/

void reporte\_materias\_semestres()

{

printf("Selecci%cn Materias por Semestre\n",162);

system("pause");

}

void menu\_reporte\_materias()

{

char opcion;

do

{

system("cls");

printf(" °----------------------------------------------------------°\n");

printf(" | ============== Sistema Acad%cmico =============== |\n",138);

printf(" °----------------------------------------------------------°\n");

printf(" °----------------------------------------------------------°\n");

printf(" ! Introduce la selección de men%c: !\n\n",163);

printf(" ! 1) Materias por Semestre !\n\n");

printf(" ! 2) Volver al men%c reportes !\n\n",163);

printf(" °----------------------------------------------------------°\n");

printf(" Opci%cn .-",162);

scanf("%c",&opcion);

switch(opcion)

{

case '1':

{

reporte\_materias\_semestres();

break;

}

case '2':

{

printf("Regresando al men%c reportes...\n",163);

system("pause");

break;

}

default:

{

printf("Opci%cn incorrecta...\n",162);

system("pause");

break;

}

}

}

while(opcion!='4');

}

int main()

{

menu\_reporte\_materias();

return 0;

}

# VII.- Code CheckList.

**Code Review Checklist**

|  |  |  |  |
| --- | --- | --- | --- |
| Student | RMM | Date | 25/04/17 |
| Program | Menú Reporte Materias | Program # | N/A |
| Instructor | PEBM | Language | C++ |

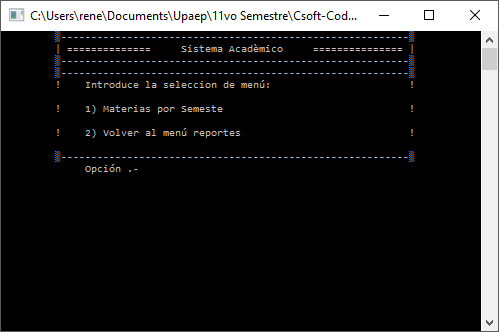
|  |  |
| --- | --- |
| Purpose | To guide you in conducting an effective code review |
| General | - Review the entire program for each checklist category; do not attempt to review for more than one category at a time!  - As you complete each review step, check off that item in the box at the right.  - Complete the checklist for one program or program unit before reviewing the next. |

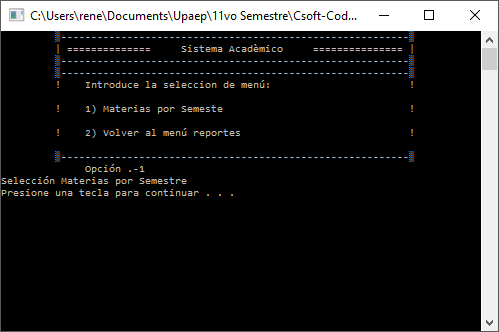
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Complete | Verify that the code covers all of the design. | MB | B | R | M |
| Includes | Verify that the includes are complete. | X |  |  |  |
| Initialization | Check variable and parameter initialization.  - at program initiation  - at start of every loop  - at class/function/procedure entry | X  X  X |  |  |  |
| Calls | Check function call formats.  - pointers  - parameters  - use of ‘&’ | X  X  X |  |  |  |
| Names | Check name spelling and use.  - Is it consistent?  - Is it within the declared scope?  - Do all structures and classes use ‘.’ reference? | X  X  X |  |  |  |
| Strings | Check that all strings are  - identified by pointers  - terminated by NULL | X  X |  |  |  |
| Pointers | Check that all  - pointers are initialized NULL  - pointers are deleted only after new  - new pointers are always deleted after use | X  X  X |  |  |  |
| Output Format | Check the output format.  - Line stepping is proper.  - Spacing is proper. | X  X |  |  |  |
| () Pairs | Ensure that () are proper and matched. |  |  |  |  |
| Logic Operators | - Verify the proper use of ==, =, ||, and so on.  - Check every logic function for (). | X  X |  |  |  |
| Line-by-line check | Check every line of code for  - instruction syntax  - proper punctuation | X  X |  |  |  |
| Standards | Ensure that the code conforms to the coding standards. | X |  |  |  |
| File Open and Close | Verify that all files are  - properly declared  - opened  - closed | X  X  X |  |  |  |

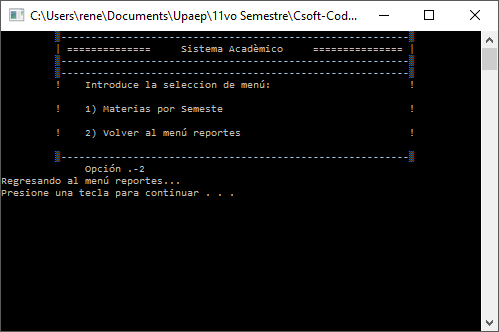
# VIII.- Reporte de resultados.

El template está vacío dado que no se especificaron pruebas para éste módulo.

Se tienen las siguientes capturas de pantalla.





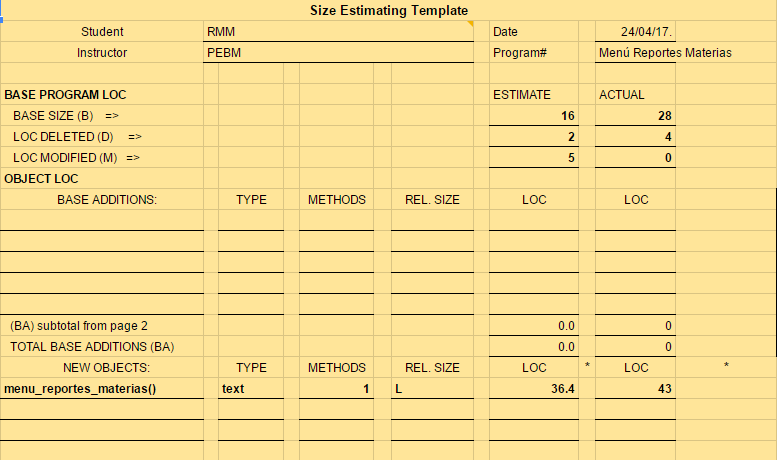


|  |  |
| --- | --- |
| Nombre de la Prueba |  |
| Caso de Uso |  |
| Objetivo de la Prueba |  |
| Descripción de la Prueba |  |
| Test |  |
| Resultados Esperados |  |
| Capturas de pantalla |  |
| Estado caso de prueba | Posibles estados:   * Ejecutado * Exitoso * Fallido * Frenado * Pendiente de ejecución * En construcción |
| Errores obtenidos |  |
| Responsable de prueba |  |
| Comentarios |  |

# IX.- PIP

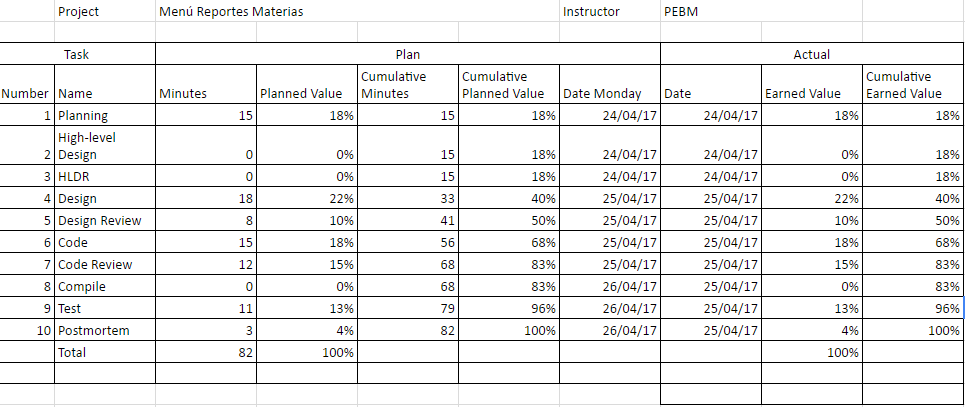
|  |  |  |
| --- | --- | --- |
| Nombre Módulo | | |
| No. PIP | Descripción del Problema | Descripción de la propuesta. |
|  |  |  |
|  |  |  |
| Nombre Módulo | | |
| No. PIP | Descripción del Problema | Descripción de la propuesta. |

# X.- Size Estimating Template



# XI.- Task and Schedule

Task



Schedule

